3/26/2014
Multiplayer Minesweeper

# Higher Order Testing

For Multiplayer Minesweeper the testing procedures we decided to perform in addition to usability testing are Stress testing and Performance Testing. The reason we chose Stress testing was to test our application with a large amount of users on our server. By doing this we can measure how our application, more specifically our server, interacts with the clients during a time when there are a large amounts users interacting. The procedure we are going to follow when Stress Testing will be creating a script to connect multitudes of people to our server. We will then test the responsiveness of our application under a heavier workload. After this we will increase the amount of connections to the server and test again. The reason that we chose Performance Testing was to check the responsiveness and stability in general under a normal workload. The way we Performance Tested our program was to have a normal amount of people on the server and then test individual components of the multiplayer gameplay. We believe that these are the best fits for our specific project because of the way that our project is set up. Stress testing made sense because of project has a specific server that connects users to each other, if the server is under high stress bugs may be present. Performance Testing makes sense in general because we want our application to run smoothly and efficiently under normal running conditions.

## Usability Testing:

### Reactions and Comments from Users

| User # | Reactions and Comments |
|---|---|
| 1 | Gameplay was fine, maybe adding a more interesting color scheme or graphics would make is nicer to look at. |
| 2 | More impressed by the multiplayer aspect and liked the chat system. Felt as though it could have been made easier/more clear as to how to connect to someone, but after doing the process once it became fairly easy. |
| 3 | Was fun and cool that you could do multiplayer! |
| 4 | I would have rather been able to select my opponent from the list on the right and then play them. |
| 5 | Did not know how to play minesweeper well, but was still impressed on the layout of our application. When moving into multiplayer |

| | |
|---|---|
| | they thought there should be more functionality in selecting an opponent. |
| 6 | Reaction was as expected, When playing single player minesweeper no certain reaction. After moving into multiplayer subject was impressed. Comments very positive, liked the chat feature and being able to see opponents moves. |

Analysis of Questionnaire

Based on the responses of the Questionnaire it seems that, for the most part, people were able to easily play both solo and multiplayer games given the instructions. The reactions were generally positive, noting the ability to play others at minesweeper as something actually fun. Other comments included things like making the UI a bit more convenient and allowing for an easier time connecting to players; as opposed to having to manually type in a very specific command each time. Overall the game seemed to please the audience and with some slight tweaking could very well be a desirable game.

Defects Found

| Defect # | Defect | How to Correct | Severity |
|---|---|---|---|
| 1 | Maximized window does not properly display the GUI | Make window non-resizeable | 2 |
| 2 | Uncover Algorithm does not uncover when clicking on the boundary of “safe block” | Rework uncover algorithm | 3 |
| 3 | Bombs left needs to be updated when you clear a section that has a “!” marked on any of the boxes | Rework the “bomb’s left” section of the uncover algorithm | 1 |
| 4 | Can keep adding flags until bomb counter goes negative | Add a check to make sure that you cannot add flags | 1 |

| | | | |
|---|---|---|---|
| | | when bomb counter is 0 | |
| 5 | Can add flags to spaces that are uncovered already | Add a check to make sure that the space to be flagged is covered | 1 |

Stress Testing:

| Workload # | Expected Result | Result | Severity + Correction |
|---|---|---|---|
| 50 Clients | Smooth and fully functional gameplay | Smooth and fully functional gameplay | No correction needed |
| 100 Clients | Smooth and fully functional gameplay | Smooth and fully functional gameplay | No correction needed |
| 200 Clients | Not as smooth but still fully functional gameplay | Very choppy gameplay | Allocate resources better |
| Over 200 Clients | Slow but functional | Not functional | Severity:3<br>Correction: Use C++ instead of java. |

Performance Testing:

| Defect # | Defect | How to Correct | Severity |
|---|---|---|---|
| 1 | When spamming the server with chat | Create a flood filter to limit incoming | 2 |

| | | | |
|---|---|---|---|
| | messages the server slows down a bit. | chat messages from a single user | |
| 2 | When opening multiple instances of the main GUI the GUI render time, especially on dual-core machines, is reduced by quite a bit | Optimize usage of local resources | 2 |
| 3 | When spamming the server with chat messages the server cannot update the chat field efficiently | Create a flood filter to limit incoming chat messages from a single user | 1 |
| 4 | When under normal workload and playing a game on two different computers with two different processing abilities there may be some “lag” when rendering opponents board | Optimize server | 3 |